ISSN 2278-3091
Volume 8, No.4, July – August 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse34842019.pdf
https://doi.org/10.30534/ijatcse/2019/34842019

Randomness Analysis on Enhanced Key Security of Playfair Cipher Algorithm

Richard M. Marzan[1], Dr. Ariel M. Sison[2], Dr. Ruji P. Medina[3]
[1]Technological Institute of the Philippines, Philippines, richard.marzan@dmmmsu-sluc.com
[2]Emilio Aguinaldo College, Philippines, ariel.sison@eac.edu.ph
[3] Technological Institute of the Philippines, Philippines, ruji.medina@tip.edu.ph

ABSTRACT

The cryptographic quality of the cipher algorithm depends on its key security. Typically, cryptographic applications provide unpredictable keys to strengthen the security of the plaintext. Despite advancements made in the field of cryptography, security loopholes still exist. As such, this paper proposed a key security of Playfair cipher algorithm by applying Playfair cipher 16x16 matrix, XOR operator, two's complement, and bit swapping. It also describes randomness tests on the proposed method that has been implemented on the existing application. We selected seven suitable randomness tests provided by NIST Test Suite. Experimental results show that the generated binary sequences are random as the proposed method passed all selected randomness tests which include frequency (Monobit) test, frequency test within a block, run test, test for the longest run of ones in a block, discrete Fourier transform, approximate entropy test, and cumulative sums test. In fact, among the different length of keys (10, 20, 30, and 40), we obtained P-values ranging from 0.01 to 1.00. It only means that no pattern has found, thus the key is unpredictable.

Key words: cryptography, key security, Playfair cipher, randomness.

1. INTRODUCTION

The exchange of different data and information over the internet demands us to increase data protection from unauthorized access or against illegal reproduction and modifications. Nowadays, the implementation of cryptography in various applications is one of the solutions. As defined, cryptography is a method of protecting data and information from any external interference so that only the intended receiving party can read and process them [1]-[3].

Among the cryptographic algorithms (e.g. Playfair Cipher, Vigenere Cipher, and Caesar Cipher), Playfair is considered as the best technique [4], [5]. It is a symmetric encryption technique that uses plaintext and translates it into a ciphertext using a key table [6], [7]. mentioned that Playfair cipher has demonstrated its success for encryption of text messages [8]. Similarly, one study affirmed that Playfair cipher is suitable for security of wireless and mobile systems since it is considered as a data encryption technique with a medium level of complexity [9].

Several studies have been proposed to enhance the key security of various cryptographic algorithms. One of which is the study of [10] where RSA algorithm uses different two keys to secure sensitive data specifically when messages are being sent over the network. In addition, hybrid cryptosystem for image file was applied to solve the problem in securing data [11]. In like manner, [12] used hybrid cryptosystem to provide high security for data communication. In their study, modified Playfair cipher 16x16 algorithm was used to encrypt and decrypt the ciphertext while Knapsack Naccache-Stern was used to encrypt and decrypt the key. Knapsack Naccache-Stern asymmetric algorithm is a deterministic public key algorithm based on knapsack problem. This algorithm has an advantage which it can produce a robust encryption result that cannot quickly to destruct. However, this algorithm has a deficiency which takes a long processing time for encryption. It has three processes namely, key generation process, encryption process, and decryption process. To generate and examine the prime number, Agrawal Kayal Saxena (AKS) algorithm, a deterministic algorithm in polynomial time was implemented.

With the problems observed in the study of [12], we were attempted to enhance the key security of Playfair cipher algorithm to address the linear proportional relationship of running time and length of cipherkey variables [13]. The results showed that the runtime performance of the proposed method outperformed the study of [12] in terms of encryption and decryption process. Also, the proposed key security algorithm manifests strong avalanche effect. It was also found out that the key cannot be easily cracked by an attacker performing brute-force attack.

To further measure the security of any cryptographic algorithms, randomness analysis is also considered [14]-[16]. Randomness is one of the security analysis to measure the cryptographic strength of a block cipher or its key. In most cases, when investigating the security of a cryptographic primitive, the objective is to find patterns between the plaintext, key, and output which do not appear uniformly. This paper describes statistical tests which measures the randomness behaviour of the key.

We present the methodology in section 2, results and conclusions are found in sections 3 and 4.

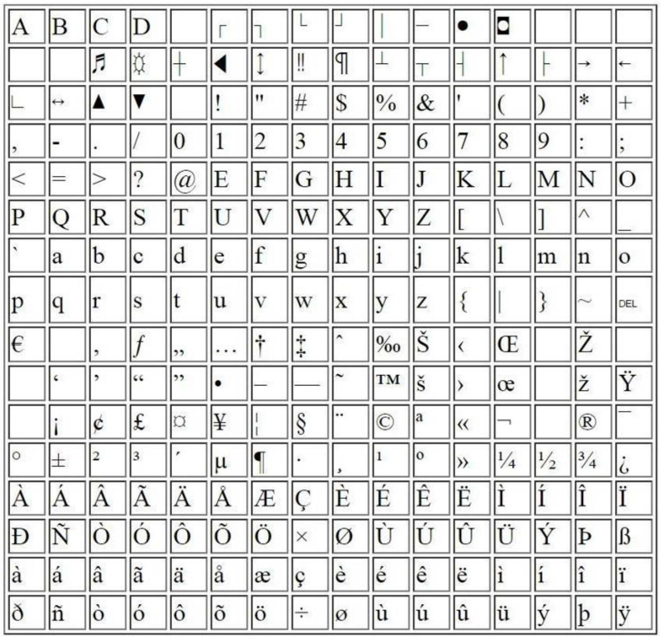
2. METHODOLOGY

This section contains the existing Playfair cipher algorithm using 16x16 matrix, the proposed method, and its security performance evaluation.

2.1 PlayFair Cipher Using 16x16 Matrix

This study used 16x16 matrix filled with numbers, alphabets (both uppercase and lowercase), and special characters as shown in Figure 1. Only these sets of characters can be combined to use as plaintext for the purpose of encrypting and producing ciphertext. To encrypt the plaintext, the following steps are implemented.

a. Read the given keyword as input.
b. Eliminate duplicated characters in the given keyword.
c. Build a table matrix by filling the characters of the keyword starting from left to right and top to bottom directions.
d. Complete the table matrix by filling with the remaining characters from ASCII values (0-255).
e. Read the plaintext as input.
f. Divide the given plaintext into pair of characters.
g. Add character "Null" when a certain character has no pair in the plaintext.



A	B	C	D		┌	┐	└	┘	│	─	●	◘			
		♬	☼	┼	◄	↕	‼	¶	┴	┬	┤	↑	├	→	←
∟	↔	▲	▼		!	"	#	$	%	&	'	(	)	*	+
,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
<	=	>	?	@	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
	‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure 1: 16x16 Playfair Matrix

On the other hand, the conversion process follows the following rules:

- If the pair of plaintext falls on the same row of the matrix table, then the letters on their right, with the first element of the row circularly following left will be their ciphertext.
- If the pair of plaintext falls on the same column of the matrix table, then the letters underneath them, with the top element of the row circularly following in the last will be their ciphertext.
- If the pair of plaintext are same, the letter on the right of the first character will be its ciphertext. The second character's ciphertext will be the letter on its left side.
- In case the pair of plaintext falls on different row and column, the letters on the same row such that they lie in column of the other letter will be their ciphertext. It must always be observed that the order is important. The first letter or character of the encrypted pair must be the first letter to be placed on the same row as the first letter of the plaintext pair.

2.2 Proposed Key Security

The proposed key security of Playfair cipher is shown in Figure 2. First, the sender will input the plaintext *PT)* together with its key *K1*. Then, the key will be used to encrypt the plaintext. To do this, the key will be arranged in a 16x16 Playfair matrix as shown Figure 1. The enhanced algorithm will be applied on the *K1* to make *CT2*. This is achieved by converting ASCII characters to decimal and the resulting output will be converted to binary numbers. This is followed by getting the two's complement of the binary numbers. The resulting binary numbers together with the key length of plain key will be XORed. Lastly, bit swapping is implemented where the last bit *l* will be in the first position, the first bit *f* will be in the second position, *l-1* will be in the third position, and so on and so forth. Sample of its process is visually illustrated in Figure 3. This completes the ciphertext of the plaintext and encryption of the key.

In doing the decryption process, the proposed algorithm will be applied on the *CT2* and find key *K1* for Playfair matrix after sending *CT1* and *CT2* to the receiver. This can be done by doing same procedures as used during encryption. However, it will be done in reverse order, i.e. bit swapping takes first and do conversion from decimal to ASCII character as the last step.

Sample encryption of the key using the proposed method applying two's complement, bit swapping, and other operations are shown in Table 1.

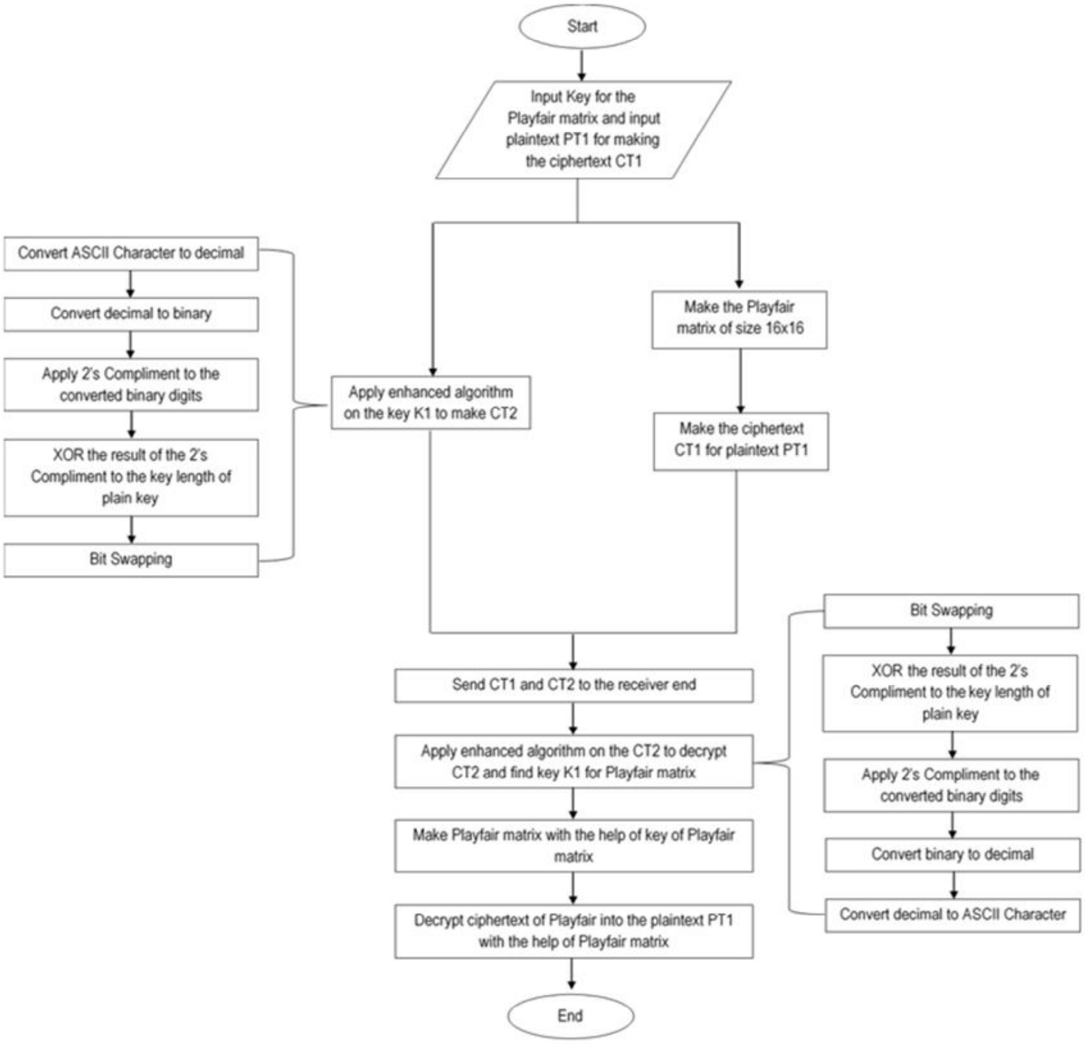

Start
Input Key for the Playfair matrix and input plaintext PT1 for making the ciphertext CT1
Convert ASCII Character to decimal
Convert decimal to binary
Apply 2's Compliment to the converted binary digits
XOR the result of the 2's Compliment to the key length of plain key
Bit Swapping
Apply enhanced algorithm on the key K1 to make CT2
Make the Playfair matrix of size 16x16
Make the ciphertext CT1 for plaintext PT1
Send CT1 and CT2 to the receiver end
Apply enhanced algorithm on the CT2 to decrypt CT2 and find key K1 for Playfair matrix
Make Playfair matrix with the help of key of Playfair matrix
Decrypt ciphertext of Playfair into the plaintext PT1 with the help of Playfair matrix
End
Bit Swapping
XOR the result of the 2's Compliment to the key length of plain key
Apply 2's Compliment to the converted binary digits
Convert binary to decimal
Convert decimal to ASCII Character

Figure 2: Flowchart of the Proposed Key Security of Playfair Cipher Algorithm

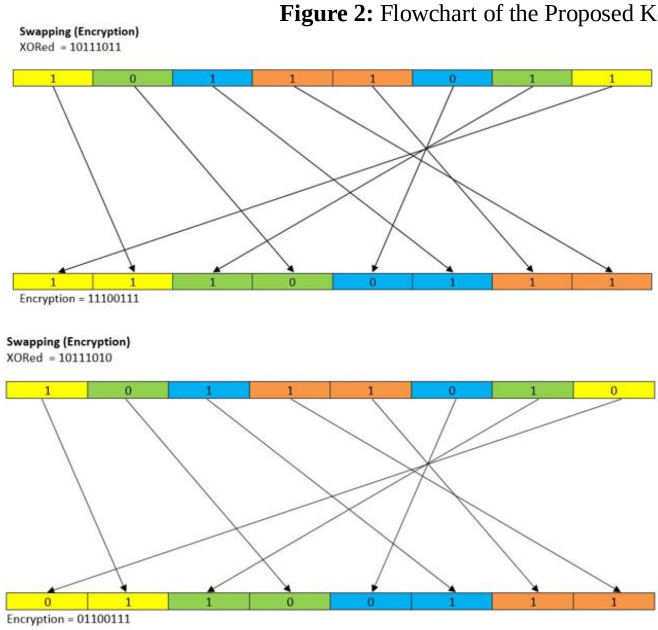


Figure 3: Sample Process of Bit Swapping

Table 1: Sample Key Encryption Using the Proposed Method

Process	Input/Output
Input plain key and plaintext	Key=ABCD, Plaintext=Hello World!
Make Playfair matrix (16x16)	see Figure 1
ASCII to decimal	A=65, B=66, C=67, D=68
Decimal to binary	65 = 0100 0001 66 = 0100 0010 67 = 0100 0011 68 = 0100 0100
Two's complement	0100 0001 = 1011 1111 0100 0010 = 1011 1110 0100 0011 = 1011 1101 0100 0100 = 1011 1100
XOR two's complement to key length=4 of plain key	1011 1111 @ 0000 0100 = 1011 1011 1011 1110 @ 0000 0100 = 1011 1010 1011 1101 @ 0000 0100 = 1011 1001 1011 1100 @ 0000 0100 = 1011 1000
Cont.	

Bit swapping	1011 1011 = 1110 0111 1011 1010 = 0110 0111 1011 1001 = 0100 0111 1011 1000 = 0100 0111
Binary to decimal	1110 0111 = 231, 0110 0111 = 103, 1100 0111 = 199, 0100 0111 = 71
ASCII code mapping	231 = ç, 103 = g, 199 = Ç, 71 = G

The integration of the enhanced key security of Playfair cipher algorithm into an application system was done after the algorithm development. The application system is the current Record Management System (RMS) of the Don Mariano Marcos Memorial State University – South La Union Campus which served as a pilot or test model in order to apply the proposed algorithm. Its functions include attaching, sending, and receiving files (e.g. communications letters, memoranda, and resolutions). These data were utilized during the performance evaluation of the proposed algorithm. All experiments were performed in Windows 10 64-bit OS Intel(R) Core (TM i3) CPU M 370 machine with 2.40 GHz of clock and 4 GB of RAM memory. Further, the proposed method was implemented in PHP.

2.3 Performance Evaluation

The security performance of the proposed key security of Playfair cipher algorithm was measured in this study. Particularly, the randomness of binary sequences of 0 and 1 produced from the proposed method was considered. In doing this, the NIST Statistical Test Suite provides 16 statistical methods [17]. However, only seven was selected as these are the most suitable in the study's main objective. These randomness tests are briefly discussed below.

2.3.1. Frequency (monobit) Test - The proportion of zeroes and ones for the entire binary sequence is the focus of frequency test. It validates whether the frequency count of zeros and ones in a binary sequence are roughly equal as would be anticipated for a truly random sequence. It further assesses how close the fraction of ones to $\frac{1}{2}$. This means that the number of zeros and ones in a binary sequence should be the same. The result of this test will serve as deciding factor to continue all successive tests. The recommended minimum input size (key) is 100 bits. The following equation is used to apply this test:

$$P - value = erfc\left(\frac{s_{obs}}{\sqrt{2}}\right) \qquad (1)$$

where,

$$erfc = \frac{2}{\sqrt{p}}\oint_{z}^{\infty} e^{-u^2}du, s_{obs} = \frac{|S_n|}{\sqrt{n}}$$

2.3.2 Frequency Test within a Block - In this test, the proportion of ones within M-bit blocks is highlighted. The purpose of this test is to determine whether the frequency of ones in an M-bit block is roughly $\frac{M}{2}$, as would be anticipated under an assumption of randomness. When M=1, this test is same with the Frequency (Monobit) test. Its recommended minimum input size for the key is 100 bits. The block size M should be selected such that $m \geq 20$ and $m > 0.01n$. Use the following equation to complete the test.

$$P - value = igamc(\frac{N}{2}, \frac{c^2(obs)}{2}) \qquad (2)$$

where, *igamc* is the incomplete gamma function for $Q(a, x)$

2.3.3 Runs Test - This test is about the total number of runs in the binary sequence, where a run is an uninterrupted sequence of identical bits. A run of length k comprises of exactly k similar bits and is bounded before and after with a bit of the opposite value. Its purpose is to ascertain whether the frequency count of runs of zeros and ones of different lengths is as anticipated for a random sequence. Specifically, the run test determines whether the oscillation between such ones and zeros is too fast or too slow. The recommended minimum input size (key) is 100 bits. Equation 3 is used to do runs test.

$$P - value = erfc\left[\frac{|V_n(obs) - 2np(1-p)|}{2\sqrt{2np(1-p)}}\right] \qquad (3)$$

2.3.4 Test for the Longest Run of Ones in a Block - This test pertains to the longest run of ones within M-bit blocks. Its main purpose is to find out whether the size of the longest run of ones within the tested sequence is uniform with the size of the longest run of ones that would be anticipated in a random sequence. It can be noted that a lack of symmetry in the anticipated length of the longest run of ones would mean that an irregularity in the anticipated length of the longest run of zeroes is present. Therefore, only a test for ones is required. Same with other tests, the recommended minimum input size (key) is 100 bits. Equation 4 is used to do this test.

$$P - value = igamc\left[\frac{K}{2}, \frac{c^2(obs)}{2}\right] \qquad (4)$$

2.3.5 Discrete Fourier Transform - The focus of this test is the peak heights in the Discrete Fourier Transform of the sequence. This test intends to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness. The purpose is to distinguish whether the number of peaks exceeding the 95% threshold is significantly different than 5%. Unlike other tests, the recommended minimum input size (key) for this test is 1000 bits. To complete this test, (5) is utilized.

$$P - value = erfc\left(\frac{|d|}{\sqrt{2}}\right) \qquad (5)$$

where,

d

2.3.5 Approximate Entropy Test- Its main attention is on the frequency of all possible overlapping m-bit patterns across the entire sequence. The approximate entropy test aims to compare the frequency of overlapping blocks of two consecutive or adjacent lengths $(m\ and\ m+1)$ against the anticipated result for a random sequence. Pick values of m and n such that $m < log_2\beta - 2$. The following equation is used to attain this test:

$$P - value = igamc(2^{m-1}, \frac{c^2}{2}) \qquad (6)$$

2.3.6 Cumulative Sums Test - This test mainly concerns to the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted $(-1, +1)$ digits in the binary sequence. It aims to see whether the cumulative sum of the partial sequences obtained in the tested sequence is too small or too large in relation to the anticipated behavior of that cumulative sum for random sequences. This cumulative sum can be regarded as a random walk. The excursions of the random walk nearing zero for a random sequence is necessary. On the other hand, the excursions of this random walk from zero will be large for certain types of non-random sequences. The recommended minimum bits for each binary sequence to be tested is 100. To attain this test, (7) is used.

$$P - value = 1 - \sum_{k=\frac{\left\lfloor \frac{n}{z}-1 \right\rfloor}{4}}^{\frac{\left\lfloor \frac{n}{z}-1 \right\rfloor}{4}} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right]$$

$$+ \sum_{k=\frac{\left\lfloor \frac{-n}{z}-3 \right\rfloor}{4}}^{\frac{\left\lfloor \frac{n}{z}-1 \right\rfloor}{4}} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right] \quad (7)$$

3. EXPERIMENTAL RESULTS AND ANALYSIS

After several experiments, Table 2 gives the results of the seven randomness tests on different key length. In this study, randomness test focus on various types of non-randomness which could exist or occur in a binary sequence generated during the encryption process. A set of P-values (corresponding to the set of sequences) is generated for each statistical test. The analysis uses a significance level α = 0.01 of the statistical tests. If the obtained P-value is ≥ 0.01, then the binary sequence of the key is random. Otherwise, it can be concluded that the 2048-bit sequences are non-random encrypted key.

Based from the table, the different key length (10, 20, 30, & 40) passed the seven selected randomness tests. The obtained P-values are ranging from 0.01 to 1.0 which is larger than the minimum significance level. It means that the proposed key security of Playfair cipher algorithm generated random binary sequence. Also, the binary sequence does not have any pattern, thus it is unpredictable. Thus, randomness tests conducted on the proposed key security of Playfair cipher algorithm increases its security against all known cryptanalytic attacks.

4. CONCLUSION AND FUTURE WORK

This study proposed a key security of Playfair cipher algorithm through encryption and decryption. The core of the algorithm was the application of 16x16 matrix and the use of XOR, two's complement, and bit swapping. After its design and development, the proposed algorithm was implemented into an existing application system. Different types of data were considered to test the proposed algorithm and eventually to evaluate its security performance using NIST Statistical Test Suite. The results show that the binary sequence generated by the key security of Playfair cipher passed all the seven statistical tests. Thus, the sequence is random and has good uniformity. The binary sequence is secured enough for secret key encryption and it is suitable to use in cryptographic applications.

Since cryptographic attack are evolving, future work should include exploring other performance evaluation metric to assess how strong the proposed method is in securing data communication and stored files. In addition, the optimization of the designed key security Playfair cipher algorithm may be carried out since it is designed to implement in software applications. Also, the proposed algorithm may consider extending to ASCII 512-character set.

ACKNOWLEDGEMENT

This work was supported by Technological Institute of the Philippines, Don Mariano Marcos Memorial State University, and K to 12 Transition Program Scholarship under Commission on Higher Education.

Table 2: Results of Randomness Tests on the Proposed Key Security of PlayFair Cipher Algorithm

Statistical Tests	**P-Value (PK=10)**	**P-Value (PK=20)**	**P-Value (PK=30)**	**P-Value (PK=40)**
1. Frequency (Monobit) Test	0.5136	0.5918	0.4354	0.0196
2. Frequency Test within a Block	0.7565	0.3104	0.4601	0.0100
3. Run Test	0.5520	0.5287	0.4837	0.0208
4. Test for the Longest Run of Ones in a Block	0.2095	0.2112	0.0213	1.6253
5. Discrete Fourier Transform	0.5236	0.6018	0.4454	0.0296
6. Approximate Entropy Test	0.2848	0.2166	0.2316	0.0603
7. Cumulative Sums Test	1.0000	1.0000	1.0000	1.0000

REFERENCES

1. S. S. Chauhan, H. Singh, and R. N. Gurjar. **Secure key exchange using RSA in extended Playfair cipher technique**, *International Journal of Computer Application*, vol. 104, no. 15, pp. 13-19, Oct. 2014. https://doi.org/10.5120/18277-9180
2. Y B. Zakariyau, Z. P. Buba, and G. M. Wajiga. **Securing message transactions through modified Playfair cipher technique**, *International Journal of Innovative Science, Engineering and Technology*, vol. 2, no. 12, pp. 760-770, Dec. 2015.
3. P. Amarendra Reddy and O. Ramesh. **Security mechanisms leveraged to overcome the effects of big data characteristics**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol 8, no. 2, pp. 312-318, 2019.
4. C. S. Subramaniyam. ***Playfair using DES algorithm, 7 by 9 matrix and colour substitution***, *International Journal of Computer Applications*, vol. 150, no. 6, pp. 30-34, Sept. 2016. https://doi.org/10.5120/ijca2016911553
5. Z. Iqbal, B. Gupta, K. K. Gola, and P. Gupta. **Enhanced the security of Playfair technique using excess 3 code (XS3) and Caesar cipher**, *International Journal of Computer Application*, vol. 103, no. 13, pp. 16-20, Oct. 2014. https://doi.org/10.5120/18134-9281
6. N. Khare and S. V. Dhari. **A survey on Playfair cipher encryption technique**, *International Journal for Scientific Research & Development*, vol. 5, no. 10, pp. 568-569, 2017.
7. A. E. Karrar and M. F. I. Fadl. **Security protocol for data transmission in cloud computing**, *International Journal of Advanced Trends in Computer Science and Engineering,* vol 7, no. 1, pp. 1-5, 2018. https://doi.org/10.30534/ijatcse/2018/01712018
8. S. Bhattacharyya, N. Chand, and S. Chakraborty. **A modified encryption technique using Playfair cipher 10 by 9 matrix with six iteration steps**, *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 3, no. 2, pp. 307-312, Feb. 2014.
9. F. Qazi, F. H. Khan, K. N. Kiani, S. Ahmed, and S. A. Khan. **Enhancing the security of communication using encryption algorithm based on ASCII values of data**, *International Journal of Security and its Application*, vol. 11, no. 3, pp. 59-68, 2017. https://doi.org/10.14257/ijsia.2017.11.2.06
10. S. K. Mathur and S. Srivastava. **Extended 16x16 Playfair algorithm for secure key exchange using RSA algorithm**, *International Journal of Scientific and Innovative Research*, vol. 5, no. 1, pp. 74-81, 2017.
11. S. M. Hardi, J. T. Tarigan, and N. Safrina. **Hybrid cryptosystem for image file using elgamal and double Playfair cipher algorithm,** *2nd International Conference on Computing and Applied Informatics 2017. IOP Conf. Series: Journal of Physics: Conf. Series 978*, pp. 1-6, 2018. https://doi.org/10.1088/1742-6596/978/1/012068
12. Amalia, M. A. Budiman, and R. Sitepu. **File text security using hybrid cryptosystem with Playfair cipher algorithm and Knapsack Naccache-Stern algorithm**, *2nd International Conference on Computing and Applied Informatics 2017. IOP Conf. Series: Journal of Physics: Conf. Series 978 (2018) 012114*, pp. 1-7, 2018. https://doi.org/10.1088/1742-6596/978/1/012114
13. R. M. Marzan and A. M. Sison. **Enhanced key security of Playfair cipher algorithm**, *in Proc 8th International Conference on Software and Computer Applications*, Penang Malaysia, 2019, pp. 457-461. https://doi.org/10.1145/3316615.3316689
14. S. Ariffin and N. A. M. Yusof. **Randomness analsis on 3D-AES block cipher**, in *Proc 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, Guilin, China, 2017, pp. 331-335. https://doi.org/10.1109/FSKD.2017.8393289
15. V. Katos. **A randomness test for block ciphers**, *Applied Mathematics and Computation*, vol. 162, no. 1, pp. 29-35, 2005. https://doi.org/10.1016/j.amc.2003.12.122
16. X. Niu, Y. Wang, and D. Wu. **A method to generate random number for cryptographic application**, in *Proc 2014 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kitakyushu, Japan, 2014, pp. 235-238. https://doi.org/10.1109/IIH-MSP.2014.65
17. L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. **A statistiacl test suite for random and pseudorandom number generators for cryptographic applications**, *(No. Special Publication (NIST SP)-800-22 Rev 1a).*